Investigation of algorithms for generating surfaces of 3D models based on an unstructured point cloud

E.S.Glumova, A.D.Filinskikh
glumova.ek@yandex.ru | alexfil@yandex.ru
Nizhny Novgorod State Technical University n a. R.E. Alekseev

Methods of 3D object model creation on the basis of unstructured (sparse) cloud of points are considered in the paper. The issues of combining point cloud compaction methods and subsequent surface generation are described. The comparative analysis of generation surfaces algorithms for the purpose of revealing of more effective method using as input data the depth maps received from the sparse cloud of points is carried out. The comparison is made by qualitative, quantitative and temporal criteria. The optimal method of 3D object model creation on the basis of unstructured (sparse) cloud of points and depth map data is chosen. The mathematical description of the point cloud compaction method on the basis of stereo-matching with application of two-phase algorithm of species search and depth map extraction from Multi-View Stereo for Community Photo Collections source image set is provided. The implementation of the method in open-source software Regard3D is realized in practice.

Key words: *3D model photogrammetry, surface generation, point cloud, depth maps.*

1. Introduction

Today, the development of computing and surface restoration technologies allows to recreate 3D models of objects with high accuracy and high quality. One of such technologies is laser scanning. With the help of laser scanners, it is possible to get the geometry of high accuracy, but unfortunately the devices that allow to achieve accuracy in hundredths of millimeters cost tens and hundreds of millions of rubles. One of the types of non-contact scanning of objects is photogrammetry. The cost of equipment for obtaining geometric data about an object is hundreds of times lower than the equipment that uses laser technology, and the main load for obtaining high-quality models falls on the software.

3D objects models are widely used in the field of parametric architecture [1], the industry of computer video games and animation [2], in the development of scenes for VR applications, as well as in mobile development [3]. The quality of the model plays an important role in any of these areas. It is important to distribute computational resources of software correctly.

There are quite a lot of various software on the market for processing images and obtaining 3D models by series of images. There are both paid software, costing about one hundred thousand rubles, and free open source software. In both cases, different algorithms are used at all stages from photo processing to obtaining a 3D model.

2. SfM Principles

One of the photogrammetry methods is the one of building a 3D structure by a set of images - Structure from Motion. The method feature is automatic determination of camera internal parameters [4]. This method restores such camera parameters as the extrinsic calibration (the orientation and position of the camera) and the intrinsic calibration (focal length, radial distortion of the lens).

The first step of SfM realization is to detect and match point features in the input images. Special points (term vary in different sources) - to put it informally - "well detectable" fragments of an image. These are points (pixels) with a characteristic (special) neighborhood - i.e. different from all neighboring points. Local features examples can be corner tops, isolated point features, contours, etc. The keypoints are described by descriptors - vectors of features computed on the basis of intensity/gradients or other characteristics of the neighborhood points.

The most popular feature descriptors used in modern image processing systems are given in [5]. A-KAZE (nonlinear diffusion filtering for detecting and describing 2D objects) is used to solve the problem of keypoint detection.

Then the camera position is assessed and a cloud of low density points or sparse points is selected. Keypoints in multiple images are matched using approximate nearest neighbor and ‘tracks’, linking specific keypoints in a set of pictures. Tracks comprising a minimum of two keypoints and three images are used for point-cloud reconstruction, with those which fail to meet these criteria being automatically discarded [6]. After that triangulation is used to estimate points three-dimensional positions and gradual reconstruction scene geometry fixed into a relative coordinate system.

An enhanced density point-cloud can be derived by implementing the Multi-View Stereo (MVS) algorithm [7], based on depth maps, the Clustering Views for Multi-View Stereo (CMVS) [8], the Patch-based MVS algorithm (PMVS2) [9], the Shadow-Aware Multi-View Stereo Algorithm (SMVS) [10], that combines stereo and shape-from-shading energies into a single optimization scheme. The camera positions obtained from a sparse point cloud are used here as input data. The result of this additional processing is a significant increase in point density.

The color and texture information is then transferred to a point cloud, after which the final 3D model is rendered.

Simplified process of obtaining 3D-model based on the images is shown in Fig. 1.

A stage of reception of surface generation on the basis of the received unstructured cloud of points by a 3D-reconstruction method MVS (Multi-View Stereo) are considered separately [7].

MVS is based on reconstructing a depth map for each view (image). Despite the large redundancy of the output data, the method has proven to be well suited for restoring the detailed geometry of sufficiently large scenes. Another advantage of depth maps as an intermediate representation is that the geometry is parameterized in its natural domain,

and per-view data (such as color) is directly available from the images. The excessive redundancy in the depth maps can cause problems; not so significant in terms of storage, but in terms of computational power [11].

MVS includes 3 stages:

- SfM, which reconstructs the parameters of the cameras;
- MVS for establishing dense;
- surface generation (meshing), which merges the MVS geometry into a globally consistent, colored mesh.

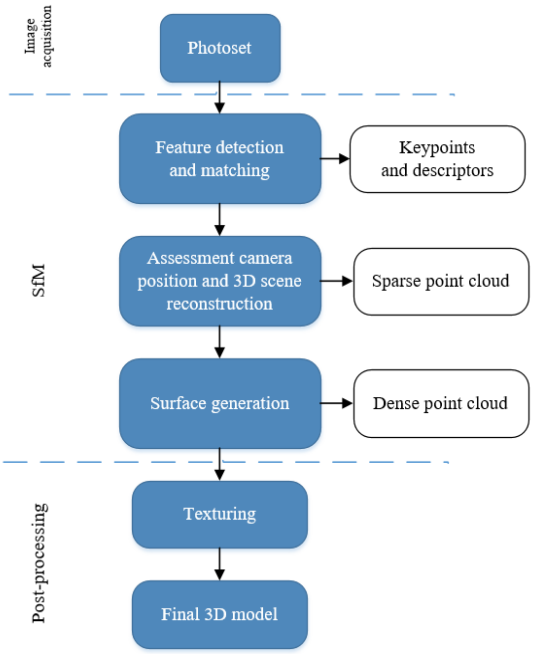

Fig. 1. Simplified process of obtaining a 3D model based on a set of images

3. Point Cloud Compression

In accordance with Fig. 1, once the camera parameters are known, dense geometry reconstruction is performed by Multi-View Stereo for Community Photo Collections (MVSCPC) [12], that reconstructs depth maps for each image. The depth map represents the two-dimensional one-channel image containing the information about distance from a sensor plane to scene objects [13].

The method is based on the idea of selecting images from the collection so that they match both per-view and per-pixel level. Appropriate choice of views ensures reliable matches even with strong differences in images. The stereo matching algorithm takes as input sparse 3D points reconstructed from SfM and iteratively grows surfaces from these points. Optimizing for surface norms with a photoconsistency measure significantly improves the matching results. The depth map quality is also assessed.

Stereo matching is performed at each pixel by optimizing for both depth and normal, starting from an initial estimate provided by a sparse point cloud. During stereo optimization, poorly matching views can be discarded and new ones added according to the local view selection criteria. The detour Pixels can be revised and their depth updated if a more accurate match is found [11].

MVSCPC provides depth map assessment for each input image - each image serves as a reference view only once, after which a two-level view selection algorithm is implemented. At the image level, global view selection determines for each reference view a set of good neighbor images to use for stereo matching.

Global view. For each reference view R, global view selection seeks a set N of neighboring views that are good candidates for stereo matching in terms of scene content, appearance, and scale. In addition, the neighboring views should provide sufficient parallax (a change in the apparent position of an object relative to a distant background, depending on the position of the observer) with respect to R and each other in order to enable a stable match. Here we describe a scoring function designed to measure the quality of each candidate neighboring view based on these desiderata.

Since matches and sparse point cloud extracted in the SfM phase are not sufficient indicators for accurate surface reconstruction (as they are extracted based on the similarity of only the scene content), another assessment of image matches reliability was proposed.

A global score g_R for each view V within a candidate neighborhood N (which includes R) as a weighted sum over features shared with R is computed as:

$$g_R(V) = \sum_{f \in F_V \cap F_R} w_N(f) \cdot w_s(f) \tag{1}$$

where F_x is the set of feature points observed in view X, and the weight functions are described below.

To encourage a good range of parallax within a neighborhood, the weight function $w_N(f)$ is defined as a product over all pairs of views in N:

$$w_N(f) = \prod_{\substack{V_i, V_j \in N \\ i \neq j,\, f \in F_{V_i} \cap F_{V_j}}} w_\alpha(f, V_i, V_j) \tag{2}$$

where $w_\alpha(f, V_i, V_j) = \min((\alpha/\alpha_{max})^2, 1)$ and α is the angle between the lines of sight from V_i and V_j to f.

The function $w_\alpha(f, V_i, V_j)$ downweights triangulation angles below α_{max}, which is usuall set to 10 degrees. The quadratic weight function serves to counteract the trend of greater numbers of features in common with decreasing angle.

The weighting function $w_s(f)$ measures similarity in resolution of images R and V at feature f. The diameter $s_V(f)$ of a sphere centered at f whose projected diameter in V equals the pixel spacing in V is computed to estimate the 3D sampling rate of V in the vicinity of the feature f.

Similarly, $s_R(f)$ is calculated for R and the scale weight w_s is defined based on the ratio $r = {s_R(f)}/{s_V(f)}$ using

$$w_s(f) = \begin{cases} {2}/{r}, 2 \le r \\ 1, 1 \le r < 2 \\ r, r < 1 \end{cases} \quad (3)$$

This weight function prefers views with equal or higher resolution than a reference view. Having defined a global estimate of species V and neighbors N, one can find the best N of a given size (usually $|N| = 10$) by the sum of species estimates $\sum_{V \in N} g_R(v)$. For efficiency, a "greedy algorithm" [14] is used and grow the neighborhood incrementally by iterative adding to N the highest scoring view, taking into account the current N (which initially contains only R).

Rescaling Views. Although global view selection algorithm tries to select neighboring views with compatible scale, some inconsistencies in scale are unavoidable due to differences in resolution within the collection of photos, which may negatively affect stereo matching. There are methods to adapt the scale of all views by filtering to a common, narrow range or global, pixel-based view. The first method is used in this research to avoid resizing of the matching window in different areas of the depth map. This approach finds a view with the lowest-resolution $V_{min} \in N$ relative to R, resamples R to approximately match that lower resolution, and then resamples higher resolution to match R.

In particular, the assessment the resolution scale of a view V relative R is based on their common features

$$scale_R(V) = \frac{1}{|F_V \cap F_R|} \sum_{f \in F_V \cap F_R} \frac{s_R(f)}{s_V(f)} \quad (4)$$

Then V_{min} simply equals $arg\ arg\ min_{\ V \in N} scale_R(V)$. If $scale_R(V)$ is less than the threshold value t ($t = 1$, which is close to the 5x5 of the reference window on a 3x3 window in the neighboring view with the lowest relative scale), the reference view is rescaled so that, after rescaling $scale_R(V) = t$. Then all neighboring views with $scale_R(V) > 2$ to match the scale of the reference view (which itself may have been changed in the previous step). It is important that all modified versions of the images are discarded when moving to the depth map computation for the next reference view.

Local View. Global view selection determines a set of N well suited candidates for a reference view and matches their scale. Instead of using all of these views for stereo matching at a specific location in the reference view, the smallest set $A \subset N$ of active views is selected (usually $|A| = 4$). Using this subset naturally speeds up the computation of the depth map.

During stereo matching, A is iteratively updated using a set of local view selection criteria designed to select views that, given a current depth and normal pixel estimates, are photometrically consistent and provide a sufficiently wide range of observation directions. To measure the photometric consistency, the mean-removed normalized cross correlation (NCC) between pixels within a window about the given pixel in R and the corresponding window in V is used. If the NCC score is above a fixed threshold, then V is a candidate for addition to A.

You can measure the angular distribution by looking at gaps of directions from which the given scene point (based on the current depth estimation for the reference pixel) is observed. In practice, the angular spread of the epipolar line [15] is considered instead, obtained by projecting each viewing ray passing through the reference point to the reference view. When deciding whether to add view V to the active set A, the local score is calculated as

$$l_R(V) = g_R(V) \cdot \prod_{V' \in A} w_e(V, V') \quad (5)$$

where $w_e(V, V') = \min({\gamma}/{\gamma_{max}}, 1)$ and γ is the acute angle between the pair of epipolar lines in the reference view as described above. Accept γ_{max}=10 degrees.

Then the local view selection algorithm is performed in the following way. Taking the initial depth of the pixel, the view V with the highest $l_R(V)$ value is found. If this view has a sufficiently high NCC score (threshold 5 is used), it is added to A; otherwise, the view is rejected. The process is repeated until either set A reaches the desired size or the view remains undecided. During stereo matching, the depth (and normal) are optimized, and a view may be removed (and marked as rejected). Then a replaced view is added. The algorithm completes as the deflected views are never revised.

4. Surface Generation

After computing arrays containing the best matching candidates for each image, you can move towards the step of surface generation. Merging the individual depth maps into a single polygonal surface is a labor intensive task. The depth maps inherit information about the multi-scale properties of the original images, which leads to vastly different sampling rates of the research surfaces.

Many approaches for depth maps fusion have been proposed [16-20]. Among them FSSR (Floating Scale Surface Reconstruction) [18] and SPSR (Screened Poisson Surface Reconstruction) [19] were considered as methods of surface generation, as they provide high detail of the reconstructed 3D model.

FSSR is widely used as outdoor scene reconstruction, when data is too sparse for a reliable reconstruction. In this case the method does not hallucinate geometry in incomplete regions, requiring manual intervention, but leaves in these areas holes (i.e. these areas have gaps).

The approach draws upon a simple yet efficient mathematical formulation to construct an implicit function as the sum of compactly supported basis functions. The implicit function has spatially continuous “floating” scale and can be readily evaluated without any preprocessing. The final surface is extracted as the zero-level set of the implicit function. One of the key properties of the

approach is that it is virtually parameter-free even for complex, mixed-scale datasets [18].

The FSSR method combines all depth maps in one large point cloud. At this stage, the scale value is attached to each point, indicating the factual size of the surface area in which the point was measured. This value is derived from the size of the regions identified in the MVS phase. Then FSSR tools calculate a multi-scale 3D surface.

SPSR is an improvement of the approach that considers surface reconstruction as a spatial Poisson problem [20]. The approach explicitly incorporates the point as interpolation constraints. Unlike other methods of image processing and geometry processing, the term screening is defined for a sparse set of points rather than for the whole area. These rare constraints, however, can be effectively integrated. Since the modified linear system retains the same finite-element discretization, the sparse structure is unchanged and the system can still be resolved using a multi-mesh approach.

In addition, Poisson's surface reconstruction presents several algorithmic improvements that together reduce the time complexity of the solution to linear in the number of points, thus enabling faster and better surface reconstruction [19].

5. Algorithm Comparison

Consider a combination of MVS-FSSR and MVS-SPSR approaches here in more detail.

Implementation is studied on the example of 21 photos of the statuette (Fig. 2) and freely distributed software Regard3D and MeshLab.

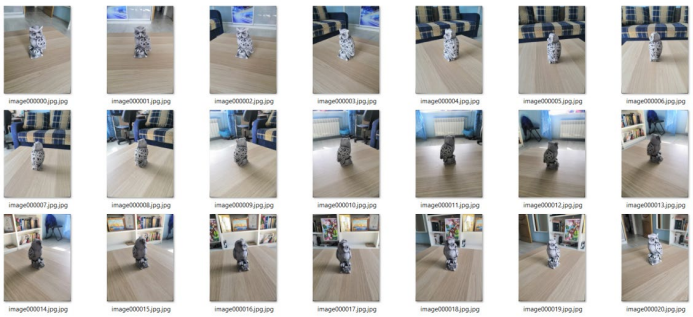

Fig. 2. Set of original photos

In Fig. 3. shows the detection of keypoints by Regard3D. This image contains 14,486 keypoints.

These key points are then matched to establish sparse matches between images (Fig. 4). The image features require the invariance to the image scaling, rotation, noise and changes in illumination.

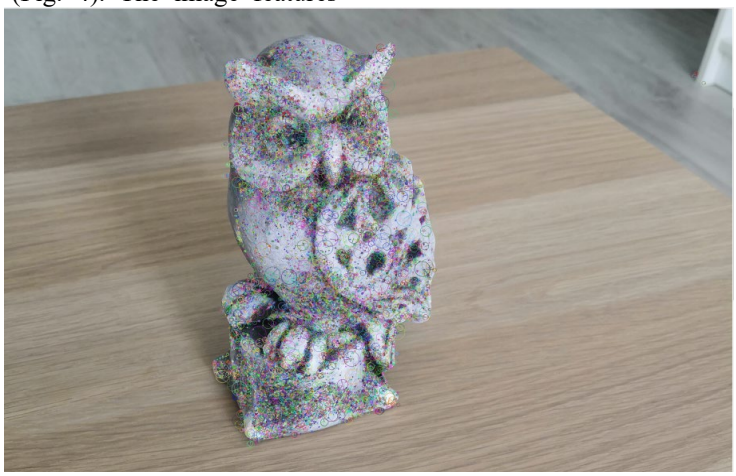

Fig. 3. Object keypoints

Fig. 4. The result of key point comparison for a pair of original images

The results of the pairing are then combined and unfolded into multiple views, creating functional tracks.

The next step in SfM implementation is incremental triangulation algorithm. It assesses the relative position of a well-matched original pair of the image, and then all tracks visible in both images are triangulated. The matching next images are incrementally added to the reconstruction until all the reconstructed views become part of the scene. Parameters of lens distortion are evaluated during the reconstruction. The performance of the following algorithms is significantly improved by removing distortions from the original images.

In Regard3D's "ideology", this method is called New Incremental. The result is a sparse point cloud (Fig. 5).

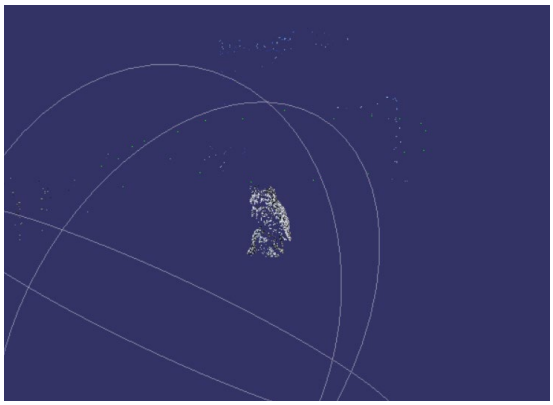

Fig. 5. The result of triangulated point cloud computing using the New Incremental method

21 cameras (according to the number of uploaded images) have been calibrated by the program, i.e. 3D positions and parameters of all images have been found. 13 583 points has received that match not only the model, but also some part of the environment. The calculation time of a point cloud has made slightly less than 30 s.

Further we will proceed to compression of the sparse point cloud by MVS method. The result of point compaction using the MVS method is shown in the figure. 6. The computation time was 40.42 minutes; 4 756 185 points were created. As you can see, the point cloud has holes on the side of the figure (Fig. 6).

The corresponding depth map was obtained using the MeshLab program (Fig.7).

Fig. 6. Dense point cloud using MVS method

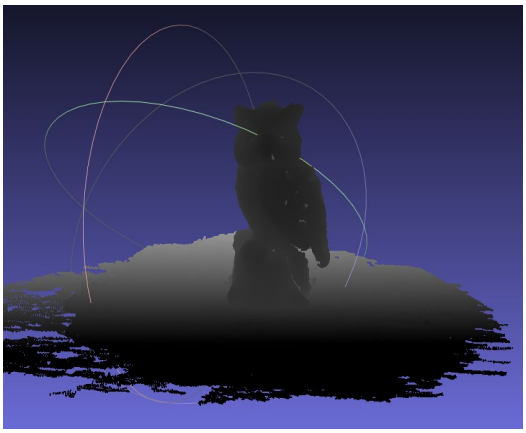

Fig. 7. Depth map

Surface generation by FSSR and SPSR. In Fig. 8. the results of calculations in the Regard3D command line are presented, illustrating the iterative algorithm of finding the best candidates for comparison described above.

```
Count:   535305  filled:   181000  Queue:     8211
Filled 79832 pixels, i.e. 2.0 %.
MVS took 147 seconds.
Count:   573874  filled:   202000  Queue:    19324
Count:    83367  filled:    37000  Queue:    20194
scaled image size: 1728 x 2304
Global View Selection: 14 15 16 18 19 20
Loading color images...
Count:   538422  filled:   182000  Queue:     8372
Processing 13583 features...
Count:   576476  filled:   203000  Queue:    19807
Processed 3906 features, from which 805 succeeded optimization.
Process queue ...
```

Fig. 8. Regard3D command line. FSSR implementation

In general, 21 reports were produced - according to the number of uploaded images. You can find the views recommended for comparison view, as well as the number of optimized points, i.e. points that have updated the depth map data and normal in accordance with the described algorithm.

Fig. 9 shows the result of FSSR method surface construction. The calculation time was 17.02 min. The final surface contains 1,369,758 points. The model also contains small noises and has gaps.

In the right picture, you can see that the model has a big hole. This is due to the fact that a shadow falls on this area in the original images. The lighting change is interpreted by the program as a lack of data for point reconstruction, because the shaded area is found in only 2-3 species out of 21, which was a rejection of its revision and surface reconstruction.

Fig. 10 shows the result of surface reconstruction using the SPSR method. The calculation time was 1.22 min. The final surface contains 301,497 points. The model also contains little noise and has gaps.

In the right picture, you can see that the model has an even greater gap than the previous method.

We will compare the obtained models by several indicators (Table 1).

Fig. 9. MVS model - Floating Scale Surface Reconstruction

Fig. 10. MVS model - Screened Poisson Surface Reconstruction

Table 1. Comparison of final models.

	FSSR	**SPSR**
Visual assessment of details	Denser mesh with lower gap area	Less dense grid with larger gap area
Calculation time	17,02 min	1,22 min
Number of points	1 369 758	301 497
Model size.obj	401 Mb	85,1 Mb

6. Conclusion

Two surface generation algorithms were considered during the research: Screened Poisson Surface Reconstruction point approach and Floating Scale Surface Reconstruction approach. In connection with the method of point compression, the considered algorithms showed different temporal and quantitative results. The result of comparison of final 3D-models generated by these methods is shown, reduction of time expenses in SPSR method does not give qualitative result. Model MVS - Screened Poisson Surface Reconstruction is a much less dense mesh than model MVS - Floating Scale Surface Reconstruction. On the basis of the received data it is possible to draw a conclusion that for reception of qualitative 3D-models on the basis of not structured point cloud it is necessary to use the algorithm of generation of the surface based on changing scale of images. The surface generation algorithm based on a point approach can be used for small collections of photos that do not contain multiscale images. Reduction of computational power in model preparation, as well as their small volume can be used, for example, for low-polygonal modeling in the mobile applications.

In the future, it is planned to conduct a comparative analysis of existing algorithms based on depth map data, as well as approaches that take into account changes in illumination in photographs.

References

[1] Sosnina, O., Filinskikh, A.; Lozhkina, N.: Analysis of the virtual nontrivial forms models creation methods (in Russian). Information technologies, T. 25. (11), pp. 679-681 (2019).

[2] Sosnina, O., Filinskikh, A., Korotaeva, A.S.: Comparison of the low-polygonal 3D model creation methods (in Russian). Information technologies, T. 23(8), pp. 564-568 (2017).

[3] Malysheva, A., Tomchinskaya, T.: Features of the low-polygon modeling and texturing in the mobile applications (in Russian). CONFERENCE KOGRAF-2019, ISBN 978-5-502-01200-3, p. 51-54.

[4] Westoby, Matthew J., et al.: Structure-from-Motion'photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology, 179, pp. 300-314 (2012).

[5] Kublanov V. and others: Biomedical signals and images in digital healthcare: storage, processing and analysis: a training manual (in Russian), pp. 193-195, (2020).

[6] Snavely K.: Scene reconstruction and visualization from internet photo collections. USA : University of Washington, (2008).

[7] Fuhrmann, Simon, Fabian Langguth, and Michael Goesele: Mve-a multi-view reconstruction environment. GCH (2014).

[8] Clustering views for multiple stereo views (CMVS), https://www.di.ens.fr/cmvs/.Last accessed 10 May 2020.

[9] Furukawa Y., Ponce J.: Accurate, Dense, and Robust Multi-View Stereopsis (PMVS). IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007).

[10] Langguth F. et al.: Shading-aware multi-view stereo. European Conference on Computer Vision, Springer, Cham, pp. 469-485 (2016).

[11] Seitz S. M. et al.: A comparison and evaluation of multi-view stereo reconstruction algorithms. 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), T.1, pp. 519-528 (2006).

[12] Goesele M. et al.: Multi-view stereo for community photo collections.2007 IEEE 11th International Conference on Computer Vision, pp. 1-8 (2007).

[13] Voronin, V, Fisunov, A., Marchuk, V., Svirin, I., Petrov, S.: Restoration of the depth map based on the combined processing of a multi-channel image (in Russian). Modern problems of science and education, 6, (2014).

[14] Greedy algorithms, https://habr.com/ru/post/120343/. Last accessed 26 June 2020.

[15] Basics of Stereo Vision, https://habr.com/ru/post/130300/. Last accessed 26 May 2020.

[16] Curless B., Levoy M.: A volumetric method for building complex models from range images. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 303-312 (1996).

[17] Fuhrmann S., Goesele M.: Fusion of depth maps with multiple scales. ACM Transactions on Graphics (TOG), T. 30 (6), pp. 1-8 (2011).

[18] Fuhrmann S., Goesele M.: Floating scale surface reconstruction. ACM Transactions on Graphics (ToG), T. 33 (4), pp. 1-11 (2014).

[19] Kazhdan M., Hoppe H.: Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG), T. 32 (3), pp. 1-13 (2013).

[20] Kazhdan M., Bolitho M., Hoppe H.: Poisson surface reconstruction. Proceedings of the fourth Eurographics symposium on Geometry processing, T. 7, (2006)

About the authors

Glumova Ekaterina S. – student of Nizhny Novgorod State Technical University n.a. R.E. Alekseev, e-mail: glumova.ek@yandex.ru.

Filinskikh Aleksandr D., Ph.D. in Technology, Associate Professor, Nizhny Novgorod State Technical University n.a. R.E. Alekseev. E-mail: alexfil@yandex.ru